Semantic Topological Map-Based Smart Wheelchair Navigation System for Low Throughput Interface

Zhixuan Wei, Weidong Chen, Jingchuan Wang and Huiyu Wang

Abstract This paper presents a new navigation system designed for the smart wheelchair, which interacts with a human user using a low throughput interface. Through the low throughput interface, the user can only give a very limited number of command types with a low frequency, meaning that a low throughput interface requests the user to do more steps, taking a longer time, just to select the desired goal. In order to decrease the number of operations requested to the user, our navigation system is designed to refine the alternative targets by selecting the significant targets and organize them in a binary tree for reducing user's operation. We also introduce a user-friendly visual feedback to display to the user in order to show the current state and the prompt commands to the user, too. This navigation system is successfully tested in a real environment.

Keywords Smart wheelchair · Semantic map · Topological map · Low throughput interface

1 Introduction

A low throughput interface requests a low operation capability to the user; therefore, it is useful for handicapped users who are not suited to control the wheelchair by the traditional interface, such as a joystick, which requires the user a more flexible and precise operation capability. A smart wheelchair normally receives two kinds of commands: speed and pose. Since a low throughput interface generates commands in low frequency and with big delays, it is difficult to control a wheelchair by speed commands via this interface. On the other hand, in an unknown environment, the

Z. Wei · W. Chen (✉) · J. Wang · H. Wang
Department of Automation, Shanghai Jiao Tong University
and Key Laboratory of System Control and Information Processing,
Ministry of Education of China, Shanghai 200240, China
e-mail: wdchen@sjtu.edu.cn

Z. Wei · W. Chen · J. Wang · H. Wang
State Key Laboratory of Robotics and System (HIT), Harbin 150001, China

E. Menegatti et al. (eds.), *Intelligent Autonomous Systems 13*,
Advances in Intelligent Systems and Computing 302,
DOI 10.1007/978-3-319-08338-4_9

smart wheelchair usually do not know the alternative target poses, and it is impossible to traverse the entire space by a low throughput interface. Hence it is not easy to control a wheelchair by pose command via a low throughput interface either. Consequently, figuring out how to control a smart wheelchair using a low throughput interface is a problem to be solved.

In recent years, many low throughput interfaces were used in order to control the smart wheelchair. Huo et al. [1] proposed a tongue–computer interface (TCI) for wheelchair control. TCI could generate six kinds of commands at 1 Hz, and used five main control commands (Forward, Backward, Turn-Right, Turn-Left, and Neutral) to control the speed of wheelchair directly. Yathunanthan et al. [2] developed a low-cost EOG signal interface to detect the EOG signal affected by eye blink, and associated the eye movement to motion commands of the wheelchair such as forward, reverse, left, and right at 1.25 Hz. However, by controlling the speed directly, the user's workload increased. Furthermore, both these two controllers could not assist the user with obstacle avoidance or navigation in order to reduce the requested user operations.

Brain–computer interface (BCI) is an important low throughput interface in smart wheelchair control. Mandel et al. [3] described the combination of a BCI interpreting steady-state visual evoked potentials and an environment analyzing system. This system analyzed a local 2D map to extract some route graphs. The user could choose a route with the four commands of the BCI. Iturrate et al. [4] proposed a brain-controlled wheelchair, which displayed to the user a real-time virtual reconstruction of the environment and the alternative targets in the traversable areas. The User could choose the desired target by EEG. Once the desired target has been chosen, the wheelchair would be controlled by an autonomous navigation system in order to move to the target while avoiding collisions with obstacles. This method controlled the smart wheelchair by the high-level commands such as move along a route or go to a position. But these goals were rough, and they did not consider the human–environment interaction.

In our previous work [5], a semantic map-based navigation system for a brain-controlled wheelchair was developed. The semantic map not only included the navigation points at the traversable areas as common navigation system, but also included the semantic target [6, 7] which could interact with human user, such as docking or door passage. The user could select the target to implement the action with only one command. Hence the operations requested to the user were reduced. However, in reference [5], a grid was used to organize the targets and users should select the goal via this grid. The navigation points at the traversable areas were too many and the grid was too complex.

In this paper, in order to further reduce the number of operations requested to the user, a semantic topological map is used. In the semantic topological map, the navigation points are refined and a new data structure is used to organize the targets. We assume that the semantic targets cover all the goals the user want, so that the function of the navigation targets is to explore the environment out of the sensor's range in order to detect more semantic targets. A binary tree is used to organize the targets and a friendly visual feedback is designed to help the user to select a target. Finally, a smart wheelchair equipped with a RGB-D sensor and BCI is developed as an experimental platform for research the effectiveness of the proposed method.

2 System Architecture

The system architecture is shown in Fig. 1. The most important part is the semantic mapping module. The wheelchair sensor information is analyzed to extract alternative targets, which represent the places where the user may be interested in. Among these targets, we have semantic targets (e.g., doors, tables) and exploration targets (such as the frontiers of sensor scan range, where the wheelchair may get more information of the environment). In order to minimize the operation of users, these alternative targets are organized by a special data structure. The user can choose a goal among the alternative targets via a low throughput interface. The motion control module will then control the wheelchair in order to reach the goal considering obstacle avoidance.

3 Semantic Mapping

The semantic topological map is the most important part in this system. In order to reduce the operations requested to the user, the system is designed to understand

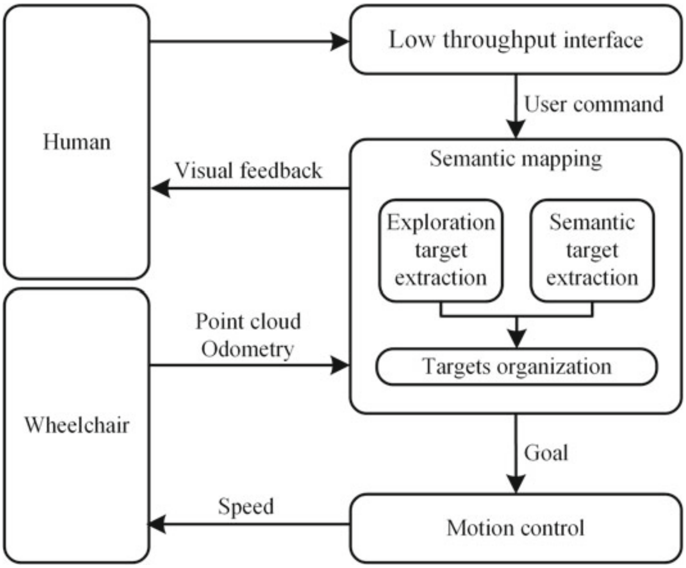


Fig. 1 System architecture

the environment and automatically find the places or the objects which the user may be interested in. The system can implement special behaviors, such as door passage or docking, along with these places or objects, called semantic targets. Since the range of the sensor is limited, the system cannot find all the semantic targets in the environment, hence the whished semantic target by the user may be out of the sensor's range. Therefore, the wheelchair needs to know where to go in order to find the semantic target outside the current range of the sensor. In order to implement this, we believe that the sensor scan range frontiers (is defined in Sect. 3.2) are an appropriate position, these positions are called exploration targets. Consequently, there are two types of targets in the semantic topological map: the semantic targets and the exploration targets.

Organizing the semantic targets and exploration targets is another problem. A good organization could help the user to select his wished target by giving a little number of commands.

3.1 Semantic Target Extraction

Referencing to our previous work [6, 7], a shape-based method is used to extract semantic target from the 3D point cloud of the environment by the following three steps:

(1) Data preprocessing: using pass through filter and down-sample to reduce the amount of data;
(2) Segmentation: using RANSAC algorithm and European clustering to segment the point cloud into horizontal planes and vertical planes;
(3) Recognition: matching the segments to a priori model library in order to identify the semantic targets.

3.2 Exploration Target Extraction

Three kinds of sensor scan frontiers are defined: leaf, border, and hopping. In some directions, the closest obstacle may be out of the sensor's rage; in these cases, the obstacle distance will be marked as the maximum distance of the measuring area of sensor. So a string of maximum distance in the sensor scan is defined as a leaf. The maximum angle and minimum angle of rays are defined as borders. The leaf and the border are important exploration targets since they mean that there are free space outside the measuring area of sensor. The last kind of frontier is hopping. If the difference between two adjacent rays is larger than a given threshold, it is defined as a hopping. A hopping means after the obstacle, there is unknown space. The middle points of each lines are defined as the exploration targets, as shown in Fig. 2.

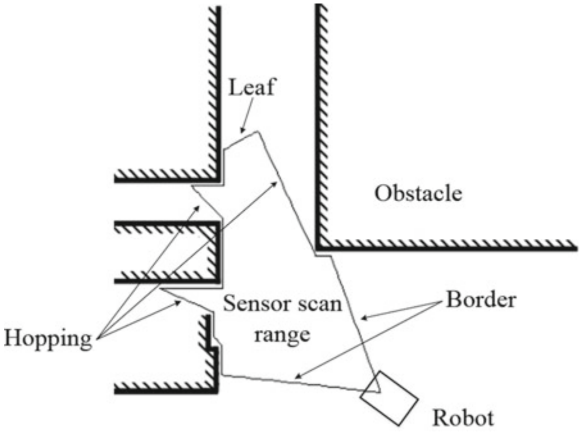


Fig. 2 Three kinds of sensor scan frontiers

3.3 Targets Organization

In order to reduce the number of operation requested to the user, the semantic targets and exploration targets are organized by a complete binary tree. The depth h of an n nodes complete binary tree is:

$$h = \lfloor \log_2 n \rfloor + 1. \quad (1)$$

Assume that low throughput interface can generate four commands: select the left child, select the right child, select the parent, and execute the current target (the current target is the goal). In an h depth binary tree, maximum steps are $h - 1$ whichever node the user wants to select with these four commands. So if number of semantic targets and exploration targets is m, maximum steps is $\lfloor \log_2 m \rfloor$ whichever target the user wants to select.

The root, the binary tree, is the current position of wheelchair. And other targets are organized by following the nearest neighbor principle. The algorithm is described by the following pseudo code:

```
allNodes = all nodes except root;
currLevel[] = { root };
{ root.childL, root.childR } = nearest two nodes;
allNodes = allNodes - { root.childL, root.childR };
currLevel[] = { root.childL, root.childR };
while(exist free nodes):
     foreach node in currLevel:
          { node.childL, node.childR } = nearest two nodes;
     foreach node in currLevel:
          allNodes = allNodes - { node.childL, node.childR };
     If node[i].child == node[j].child:
          If dis(node[i], node[i].child) > dis(node[j], node[j].child):
               node[i].child = nestest node;
               allNodes = allNodes - { node[i].child };
          else:
               node[j].child = nestest node;
               allNodes = allNodes - { node[j].child };
```

Consequence, the semantic topological map is expressed by the binary tree.

3.4 Visual Feedback

The visual feedback, which directly interacts with the user, should be simple, easy to understand, and easy to use. Besides the semantic topological map, the 3D map of the environment is also displayed on the screen in order to show the position of the targets in the real world. In order to help the user to select the wished target, four colors (Blue, green, red, and yellow) are used to indicate four commands. Regardless of which target the user wants to select, what he needs to do is just issue the command linked to the current color assigned to the target.

For example in the Fig. 5, these are seven targets in the semantic topological map. Each ball indicates a target, except the bottom one (the root). The white rectangle indicate a semantic object. At the beginning (Fig. 5c), no target is selected, and the user can use either the red or the yellow command to turn the wheelchair to the left or to the right. If the user wants to select right door in Fig. 5c, since the target is green right now, the user can issue green command first. Than the target turn to blue (Fig. 5d), and the user can issue blue the command now, and so on. It is very easy to use and understand.

4 Motion Control

The wheelchair is controlled by a real-time feedback controller [8]. For the motion control the exploration and semantic targets are slightly different: the semantic targets are poses, which include position and orientation, while the exploration targets are poses that include just a position. For this reason, the controller can behave in two modes.

If the goal is a semantic target, the linear and angular speed (ν and ω) are calculated according to the relative position between the wheelchair and the goal (Eq. 2):

$$\begin{cases} \mathrm{v} = k_r r \\ \omega = k_a a + k_b b \\ \mathrm{r} = \sqrt{\Delta x^2 + \Delta y^2} \\ a = -\theta + \mathrm{atan2}(\Delta x, \Delta y) \\ b = -\theta - a \end{cases} . \tag{2}$$

where Δx and Δy are position error between the wheelchair and the goal, θ is the orientation of wheelchair, r is the distance between the wheelchair and the goal, a and b are intermediate variables. k_r, k_a and k_b are constants.

If the goal is an exploration target, the algorithm is a simplification of Eq. 2 as shown in Eq. 3.

$$\begin{cases} \mathrm{v} = k_r r \\ \omega = k_a \mathrm{atan2}(\Delta x, \Delta y). \\ \mathrm{r} = \sqrt{\Delta x^2 + \Delta y^2} \end{cases} \tag{3}$$

where the meanings of all parameters are similar with Eq. 2.

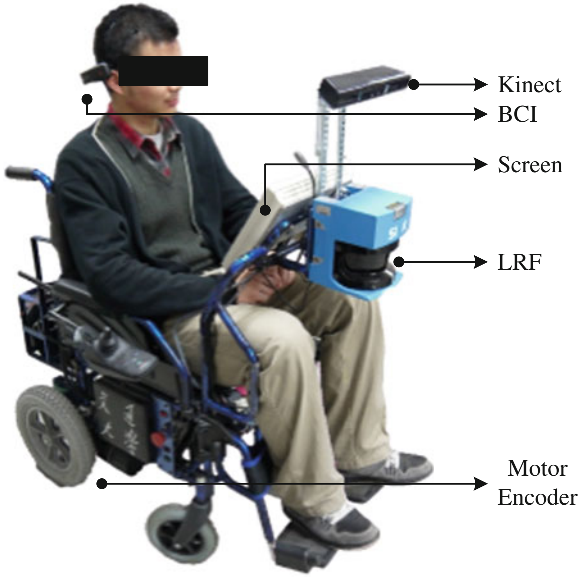
Meanwhile, the obstacle avoidance algorithm MVFH&VFF [9, 10] is implemented to guarantee the safety. The obstacles are extracted from the 3D point cloud. For more detail, please refer to our previous work [5].

5 Experiment and Results

5.1 Wheelchair Prototype and Experiment

The wheelchair prototype [6, 7] as shown in Fig. 3 equipped with several mobile robot sensors including Kinect, odometry, Emotiv EPOC Neuroheadset, etc., is a customized model of an ordinary electric wheelchair (LRF in the Fig. 3 is not used in this paper). Two computers are mounted on the wheelchair. Semantic mapping module and motion control module are running a Linux computer while a Windows computer for BCI sends the commands to the Linux computer via Ethernet. A smart motion controller (SMC) communicates with the computer via serial and is used to control the speed of the wheelchair at 20 Hz. The 3D point cloud is obtained from Kinect for semantic targets extraction and obstacle avoidance. And the 3D point cloud between the height of 1.5–0.1 m is projected to the ground in order to generate fake laser scan for exploration extraction. The semantic topological map update frequency is 1 Hz. The EPOC is connected to the computer by Bluetooth for obtaining BCI commands in 3 Hz. The system software is developed based on ROS [11] and PCL [12].

Fig. 3 Wheelchair prototype


[redacted]

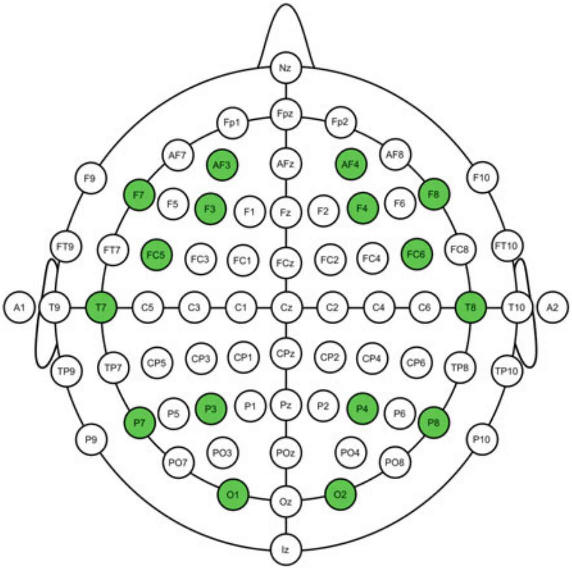


Fig. 4 Electrodes of International 10–20 system for EEG [13]

The environment for experiment is shown in Fig. 6a. The tasks include passing through a doorway and docking into the table.

5.2 Brain–Computer Interface

Emotiv EPOC is a wireless neuroheadset using a set of 14 sensors plus 2 references. The 14 channels of the EPOC distribute in accordance with International 10–20 system [14] (as shown in Fig. 4) locations corresponding to AF3, AF4, F3, F4, F7, F8, FC5, FC6, P7, P8, T7, T8, O1, O2, as well as 2 references channels are positioned at P3 and P4. The Expressiv^{TM} Suite is a component of Emotiv EPOC. It uses the signals (including EEG and EMG) measured by the neuroheadset to interpret player facial expressions in real time. We define four expressions as commands: lifting brows, biting the teeth on the left, biting the teeth on the right, and biting the teeth on both side. These expressions are sensitive, and do not affect the user operating the wheelchair (e.g., moving eyes will affect the user observing screen).

5.3 Semantic Mapping

Figure 5 illuminates an example of semantic mapping test. Figure 5a is a photo of experimental environment, and Fig. 5b is a sketch showing the environment and the pose of wheelchair. Figure 5c–f show the process for selecting a target. From Fig. 5c–f, the user sends the commands in turn: select the right child (c to d), select the left child (d to e), and execute the current target (e to f).

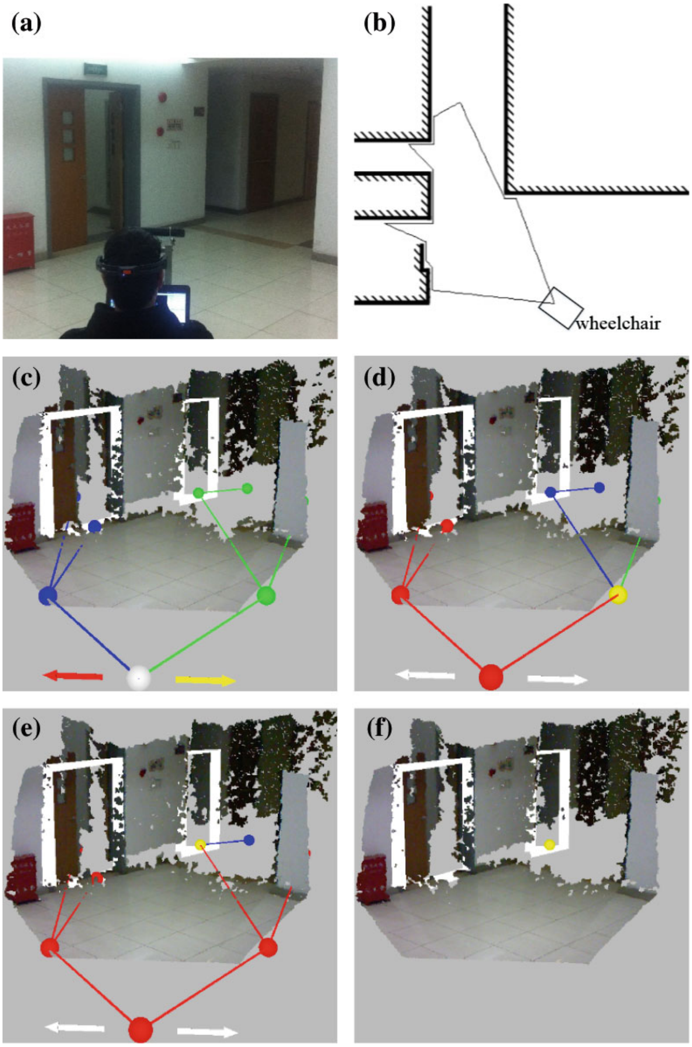


Fig. 5 Semantic mapping test

5.4 Discussion

In this section, we compare our current system with our previous system [5], and evaluate the performance of these two systems. The evaluations are described in our previous work [5].

Figure 6b shows the comparison of trajectories two systems and the results are summarized in Tables 1, 2, and 3. All our five experiments on both systems succeeded. In the current system, the wheelchair will stop at each goal and wait for the user new

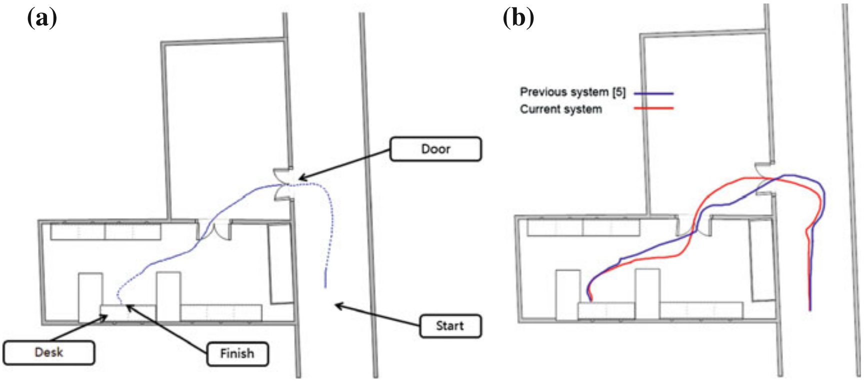


Fig. 6 **a** Experimental environment and tasks; **b** Comparative experiment

Table 1 Metrics to evaluate the wheelchair performance

	Previous system [5]		Current system	
	Mean	Std.	Mean	Std.
Task success	1	–	1	–
Path length (m)	28.5	0.3	25.3	0.3
Time (s)	96	8	105	7

Table 2 Metrics to evaluate the comfortability

	Previous system [5]		Current system	
	Mean	Std.	Mean	Std.
High ω ratio (%)	3.4	1.1	3.3	1.2
BCI commands	19	2	15	1

Table 3 Metrics to evaluate the safety

	Previous system [5]		Current system	
	Mean	Std.	Mean	Std.
Collisions	0	0	0	0
Clearance mean (m)	1.56	0.03	1.55	0.03
Clearance min (m)	0.65	0.07	0.66	0.06

command; hence, it takes more time to control the wheelchair and it also results in a less smooth final trajectory of our system.

It is obvious that the BCI commands to give in the current system are reduced. The user only needs to send commands to the navigation system when the wheelchair stops at the last goal and is waiting for a new goal.

However, there is still room for improvement. If the user changes his mind while selecting a target, he may need more steps to select another target in the semantic topological map. This is determined by the nature of the binary tree. Furthermore, if the desired target is not extracted by the system, then the navigation system can be switch to the speed control mode (similar as our previous work in [5]) in order to arrive any position.

6 Conclusion and Future Works

This paper presents a smart wheelchair navigation system relying on semantic topological map controlled via low throughput interface. Semantic topological map consists in a complete binary tree composed of exploration and semantic targets. With a friendly visual feedback, the user can easily select the desired target. With the help of the proposed navigation system, the number of operations requested to the user are obviously reduced. However, the ability of the semantic extraction module will influence the performance of proposed navigation system. Hence the robustness and stability of semantic mapping should be a priority, and the kinds of objects recognizable as targets must be expanded.

Acknowledgments This work is partly supported by the National High Technology Research and Development Program of China under grant 2012AA041403, the Natural Science Foundation of China under grant 60934006 and 61175088.

References

1. Huo, X., Ghovanloo, M.: Evaluation of a Wireless Wearable Tongue-Computer Interface by Individuals with High-Level Spinal Cord Injuries. Journal of Neural Engineering 7, 026008 (2010).
2. Yathunanthan, S., Chandrasena, L. U. R., Umakanthan, A., Vasuki, V., Munasinghe, S. R.: Controlling a Wheelchair by Use of EOG Signal. In: Information and Automation for Sustainability, 2008. ICIAFS 2008. 4th International Conference on, pp. 283–288. (2008).
3. Mandel, C., Luth, T., Laue, T., Rofer, T., Graser, A., Krieg-Bruckner, B.: Navigating a Smart Wheelchair with a Brain-Computer Interface Interpreting Steady-State Visual Evoked Potentials. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1118–1125. IEEE Press, St. Louis (2009).
4. Iturrate, I., Antelis, J. M., Kubler, A., Minguez, J.: A Noninvasive Brain-Actuated Wheelchair Based on a P300 Neurophysiological Protocol and Automated Navigation. IEEE Transactions on Robotics 25, 614–627 (2009).
5. Wei, Z., Chen, W., Wang, J., Wang, H., Li, K.: Semantic Mapping for Safe and Comfortable Navigation of a Brain-Controlled Wheelchair. In: Su, C.-Y., Rakheja, S., and Liu, H. H. (eds.) Intelligent Robotics and Applications. LNCS, vol. 8102, pp. 307–317. Springer, Heidelberg (2013).
6. Wei, Z., Chen, W., Wang, J.: 3D Semantic Map-Based Shared Control for Smart Wheelchair. In: Su, C.-Y., Rakheja, S., and Liu, H. H. (eds.) Intelligent Robotics and Applications. LNCS, vol. 7507, pp. 41–51. Springer, Heidelberg (2012).

7. Wei, Z., Chen, W., Wang, J.: Semantic Mapping for Smart Wheelchairs Using RGB-D Camera. Journal of Medical Imaging and Health Informatics 3, 94–100 (2013).
8. Siegwart, R., Nourbakhsh, I. R.: Introduction to Autonomous Mobile Robots. The MIT Press, Cambridge (2004).
9. Parikh, S. P., Grassi Jr, V., Kumar, V., Okamoto Jr, J.: Integrating Human Inputs with Autonomous Behaviors on an Intelligent Wheelchair Platform. IEEE Intelligent Systems 22, 33–41 (2007).
10. Carlson, T., Demiris, Y.: Human-Wheelchair Collaboration through Prediction of Intention and Adaptive Assistance. In: IEEE International Conference on Robotics and Automation, pp. 3926–3931. IEEE Press, Pasadena (2008).
11. Robot Operating System (ROS). http://www.ros.org
12. Rusu, R. B., Cousins, S.: 3D is Here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation, pp. 1–4. IEEE Press, Shanghai (2011).
13. International 10–20 system. http://en.wikipedia.org/wiki/10-20_system_(EEG)
14. Niedermeyer, E., da Silva, F. H. L.: Electroencephalography: Basic Principles, Clinical Applications, and Related Fields. Lippincott Williams & Wilkins, Hagerstown (2005).